

Improving SAS® Performance by Improving I/O Throughput of SAS® Work Directory

Lu Xu, Procter & Gamble Pharmaceuticals, Cincinnati, OH

Abstract

Each SAS data set has its corresponding file in the SAS work directory. Creating or reading a SAS data set involves I/O activities in the SAS work directory. As CPU speed has increased dramatically for the past decade and multiple CPUs are available on a system, the I/O throughput of the SAS work directory could easily become a system bottleneck and a key factor in determining SAS performance. Using a HP 9000 UNIX system as an example, this paper first shows how seriously SAS performance can be affected by this I/O bottleneck. Then it discusses using different system configurations and storage hardware to improve SAS performance. It shows that SAS performance can be tripled using a fast storage device for the SAS work directory and a modified system configuration. This method can also be applied to other computer platforms to improve SAS performance. This paper is for those who want to improve their SAS performance, i.e., run their SAS jobs faster. No particular level of SAS expertise is required to benefit from this paper.

Introduction

In early 1996, we bought a HP 9000 T500 server system with seven CPUs, 1GB RAM system memory and 2MB cache to replace a VAX system. At that time we were running the system under the HP UNIX operating system HP-UX 10.01. The primary application on this system is SAS which is used for statistical analysis. Considering the CPU speed and other system components, the HP system should execute SAS programs much faster than the exiting VAX system. However, during the testing period we found it took much longer time than expected to execute some SAS programs. To get a benchmark of SAS performance, seven typical SAS programs used by the statisticians were chosen and run simultaneously when there were no other programs running on the system. The benchmark result showed that the performance was far below the expectation and would not meet our computing needs. This paper presents the problem of low SAS performance caused by I/O bottleneck and the solution to this problem.

Cause of the Problem

The I/O intensive SAS programs (used for Monte Carlo simulation) which contained many SAS data step iterations took more time to finish than expected. Using Glance Plus, a system performance monitor for the HP-UX, it was found that the low SAS performance was caused by an I/O bottleneck on the file system where the SAS work directory resided. Figure 1 shows the disk I/O activities with and without running two simulation programs. The utilization of the disk where the SAS work directory resided reached 100% when the two simulation programs were running.

Each SAS data set has a corresponding file in the SAS work directory. Creating or reading a SAS data set involves I/O activities in the SAS work directory. These simulation programs created many SAS data sets (or the same data set name but many times) which resulted in an I/O bottleneck on the SAS work directory. Therefore the I/O throughput of the SAS work directory was determined to have a significant impact on the execution time of I/O intensive SAS programs such as simulation programs.

For the HP9000 system, the SAS work directory has to be created on a UNIX file system which can only be created on disks, not within the system memory (RAM). Compared to CPU speed, it takes more time to do read/write to a disk because of disk seek time and physical read/write time. This explains why the I/O bottleneck occurred on the SAS work directory.

Solution

We used a combination of reconfiguring the I/O system and adding new storage hardware to improve I/O throughput of the SAS work directory file system to improve the SAS performance. Another technique which might be helpful would be to use several SAS work directories created on different disks to reduce I/O activities on a single disk. *SAS Companion for UNIX Environments: Language*¹ also suggests some other ways to improve the performance.

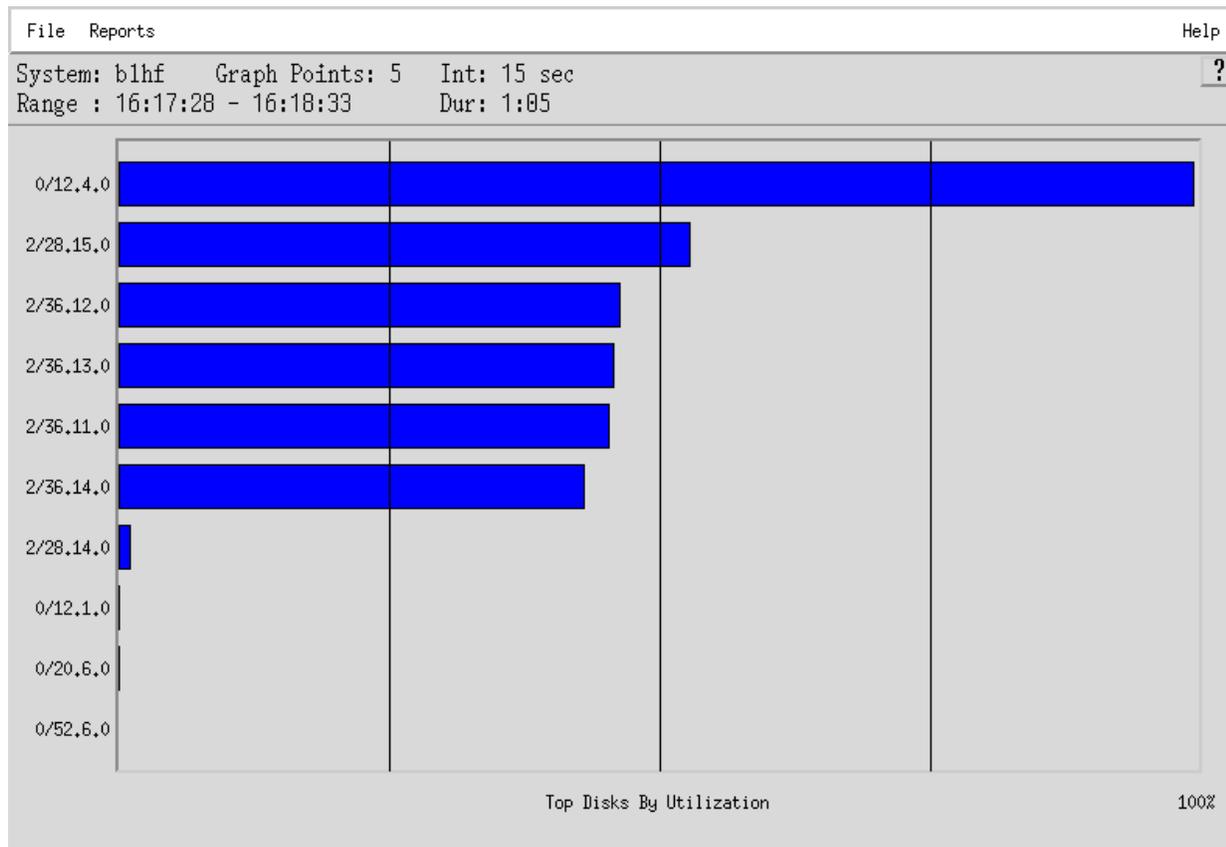
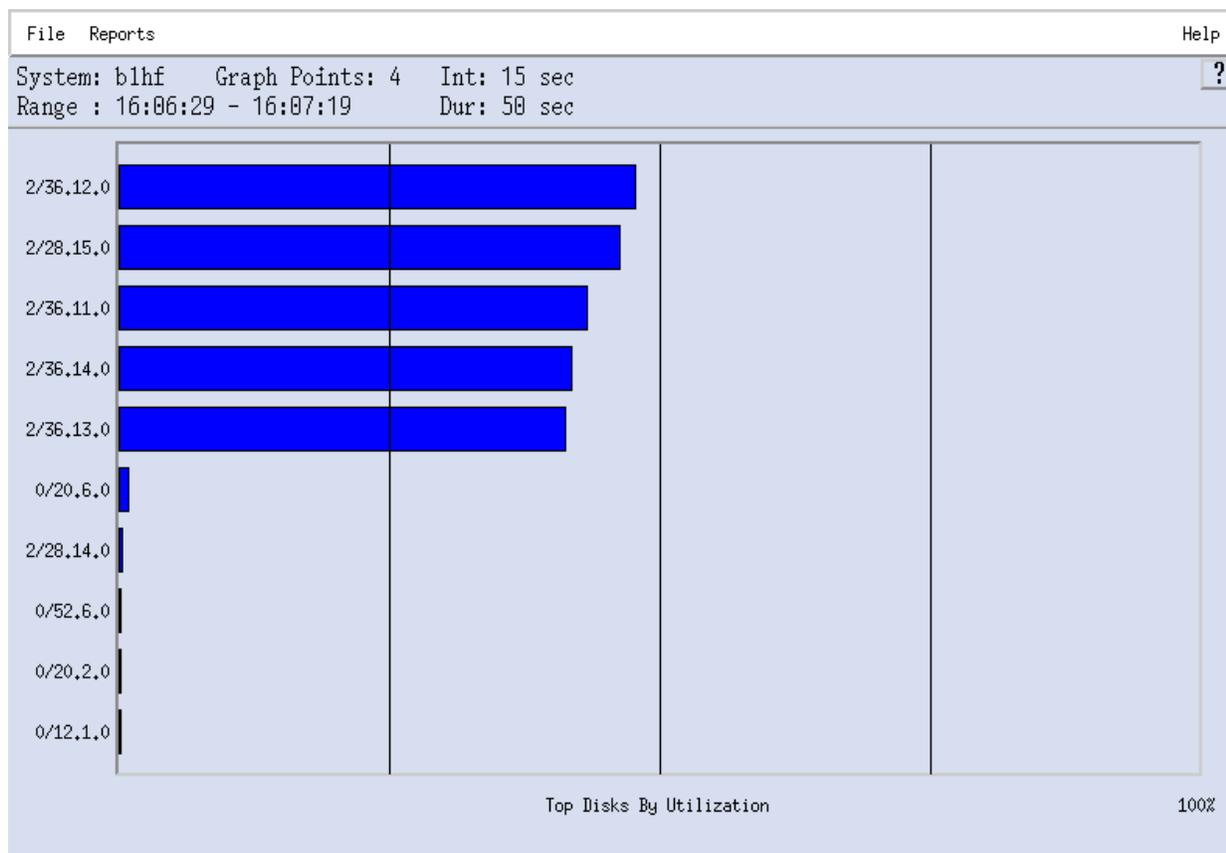


Figure 1. Disk Activities without (Top) and with (Bottom) Running Two Simulation Programs. SAS work directory is on disk 0/12.4.0 Vertical Axis: Disk Name Horizontal Axis: Disk Utilization

Reconfiguring the I/O System

1) *Using a dedicated disk controller*

The speed of read/write access to a disk is not only determined by the read/write speed of the disk itself but also the speed that data can go through the disk controller. Usually a disk controller controls 6 to 8 disks. Some other file systems, such as those for users' home directories, may share the same disk controller with the SAS work directory file system. The I/O throughput of the SAS work directory can be reduced if other file systems have heavy I/O activities. We decided to use a dedicated disk controller just for the SAS work directory file system to achieve better I/O performance.

2) *Using disk striping*

Disk striping is bundling several disks together to read/write information from/to these disks in parallel to achieve higher I/O throughput. Considering the available resources, the size of the SAS work directory and other factors, we used four 2GB disks as a disk set for disk striping, and created a file system on it for the SAS work directory.

3) *Choosing the optimal block size for the file system*

When system reads/writes data from/to disks, it reads/writes the data in blocks. Depending on the nature of the SAS programs, the block size of the file system has an effect on the I/O throughput. In order to find the optimal block size for our typical work, a test program set was run against file systems with different block sizes.

4) *Testing synchronous vs. asynchronous disk I/O*

Synchronous disk I/O means that the writing process does not consider the write complete until the physical write to the disk platter has completed. While asynchronous disk I/O means that the writing process considers the write complete after the data has been written to the cache on the disk -- leaving the physical write to the disk platter to the disk itself. Synchronous disk I/O is standard on the HP-UX. Although changing from synchronous to asynchronous was not expected to make a difference on SAS performance, it was tested.

Adding New Storage Hardware

We bought a 512MB solid state disk which is made of memory arrays and functions like a disk. The

throughput (data rate) of this solid state disk is 20 MB/sec which is much faster than that of the disks on the HP. Because it is made of RAM, the solid state disk needs no seek time and has no latency. Therefore, it can provide high I/O performance. The solid state disk is commonly used with databases, for example, Oracle, to achieve high database application performance.

Tests and Results

Typically many programs would run simultaneously on all seven CPUs when the system was used for production. Seven typical SAS programs used by the statisticians were chosen to make a test program set. It included both CPU and I/O intensive SAS programs. To find which combination of the techniques would give the best SAS performance, the test programs were run against different system setups when there were no other system and user activities. The performance was defined as the time needed to complete all seven programs.

Table 1 summarizes the test results for one of the test programs under different system setups. The test results of the whole test program set are summarized in Table 2. A dedicated disk controller was used for all the system setups.

Discussion

The results in Table 1 are for a moderate I/O intensive SAS program. Total execution time is separated into CPU time and non-CPU time, where the non-CPU time is mainly the I/O time. Using the three system setups, the CPU time remained quite stable. However, when switching from setup 1 to setup 2, the non-CPU time decreased sharply from 33.03 minutes to 3.40 minutes, a 90% reduction. In terms of total execution time, setup 2 needed only one quarter of the time that setup 1 needed. For all the seven programs in the test set, the non-CPU time decreased about 90% with slight CPU time increase (<4%) when switching from setup 1 to setup 2. For all seven programs, setup 2 was 2.2 to 4.7 times faster than setup 1. When switching from setup 1 to setup 3, for all seven programs, the non-CPU time decreased about 50% while the CPU time increased by 4% to 16%. Overall, setup 3 was 1.4 to 1.7 times faster than setup 1 for the seven programs.

The results in Table 2 show that the SAS performance of using the solid state disk was 2.5 to 2.8 times faster than the benchmark while the performance of using the disk striping was 1.2 to 1.5 times faster than the

benchmark. The optimal file system block size was 8K for the solid state disk and 4K for the disk striping. Synchronous or Asynchronous disk access did not make much, if any, difference on performance as

expected. Overall the solid state disk with an 8K file system block size gave the best SAS performance. This setup almost tripled the benchmark performance and doubled the performance of the disk striping.

System Setup	Total Execution Time (minutes)	CPU Time (minutes)	Non-CPU Time (minutes)
1. Disk without Disk Striping, 8K Block Size and Synchronous I/O	39.50	6.47	33.03
2. Solid State Disk, 8K Block Size and Synchronous I/O	9.95	6.55	3.40
3. Disk with Disk Striping, 4K Block Size and Synchronous I/O	24.32	7.15	17.17

Table 1. Test Results for One Test Program Under Different System Setups

Hardware	File System Block Size	Synchronous/Asynchronous	Performance ⁺ (relative to benchmark)
Benchmark*	8 K	Synchronous	100
Solid State Disk	8K	Synchronous	282
Solid State Disk	4K	Synchronous	251
Solid State Disk	16K	Synchronous	251
Solid State Disk	8K	Asynchronous	283
Disk Striping	8K	Synchronous	149
Disk Striping	32K	Synchronous	118
Disk Striping	4K	Synchronous	151
Disk Striping	4K	Asynchronous	151

+ Performance = (Time for benchmark)/(Time for system setup) x 100.

* Benchmark setup was disk without disk striping.

Table 2. Test Results of the Whole Test Program Set Under Different System Setups

Final SAS Work Directory Setup

On our system, the solid state disk with a dedicated disk controller is used solely for one of the two SAS work directories. This directory is shared by all SAS programs except those which need larger space than that of the solid state disk. Programs which need large space use the other SAS work directory with 4GB space (the maximum file system size under HP-UX10.01). This directory was created on a four 2GB disk set using disk striping. This dual SAS work

directory setup has been running for more than one year.

During this time our experience has shown this setup to be very robust. At one time there were six simulation programs along with other programs running and the SAS performance was still very satisfactory.

Conclusion

I/O bottleneck could be the cause of low SAS performance. Reconfiguring I/O system and/or using high performance disks such as solid state disk for SAS work directory can greatly improve the SAS performance.

Acknowledgements

The author would like to thank John Taulbee and Charles Taylor for their careful review of the manuscript and their support for the project. The author would also like to thank Mohammad Hoseyni, Denise Jeavons and David Loecke for their careful review of the manuscript.

References

1. SAS Institute Inc., SAS[®] Companion for UNIX Environments: Language
2. SAS Institute Inc., SAS[®] Language

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Lu Xu
Procter & Gamble Pharmaceuticals
Sharon Woods Technical Center
11450 Grooms Road
Cincinnati, OH 45242
(513) 626-5579
xu.l@pg.com